

**AMENDMENTS TO THE CLAIMS**

1. (Currently Amended) A method for controlling model execution in a graphical modeling environment, the method comprising:

displaying a view of a graphical model with a plurality of executable time-based components, said graphical model including at least one user-configurable graphical post component representing an event, said graphical post component having at least one input port for receiving at least one input signal, said graphical post component specifying a condition associated with the at least one input signal ~~that is satisfied during execution of said graphical model before a posting of a notice of said event occurs;~~

identifying when said condition is satisfied during execution of said graphical model;  
posting said notice of an occurrence of said event in said graphical modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event; and

executing at least one component from said plurality of components in response to said notifying.

2. (Previously Presented) The method of claim 1, further comprising:

registering at least one of said plurality of components with said event handler; and  
receiving at said at least one of said plurality of components registering with said event handler, notification of the occurrence of said event following said posting.

3. (Previously Presented) The method of claim 1 wherein the graphical post component is a block or label.

4. (Currently Amended) The method of claim 1, further comprising:

setting a sample time for an initial execution of at least one component to be the occurrence of the specified event;

5. (Previously Presented) The method of claim 4, further comprising:

propagating the sample time to at least one other component in said graphical model, said at least one other component configured to inherit a sample rate.

6. (Previously Presented) The method of claim 4, further comprising:  
    setting a sample time of a plurality of non-contiguous components in said graphical model to be the occurrence of said event.
7. (Original) The method of claim 6 wherein said sample time for the plurality of non-contiguous components is set without adjusting visible connections between components displayed in said view.
8. (Previously Presented) The method of claim 4, further comprising:  
    indicating with an event ID in said view that the sample time of said at least one component is set to said event.
9. (Original) The method of claim 4 wherein said event is an explicit event set by a user.
10. (Previously Presented) The method of claim 4 wherein said event is an implicit event caused by the execution of the graphical model.
11. (Original) The method of claim 10 wherein the implicit event is one of power-up, power-down and initialization.
12. (Original) The method of claim 10 wherein the implicit event corresponds to one of the enabling and disabling of a subsystem.
13. (Previously Presented) The method of claim 2, further comprising:  
    indicating which event a component receives with a user-configurable color in said view.
14. (Previously Presented) The method of claim 1, wherein an execution scope of the specified event for which the execution of the graphical model is being monitored is restricted to a portion of the graphical model.
15. (Previously Presented) The method of claim 1 wherein each event in said graphical model maps on a one-to-one basis to an event handler, said event handler being a function.

16. (Original) The method of claim 15 wherein said function is inlined.

17. (Original) The method of claim 1 wherein a branch priority block indicates an order of execution among at least two branches of blocks in response to said notifying.

18. (Original) The method of claim 1 wherein more than one block group executes in response to said notifying, said block groups being a user selected grouping of blocks, the order of execution of the block groups specified by a user.

19. (Currently Amended) A method for controlling model execution in a modeling environment, the method comprising:

displaying a view of a model with a plurality of executable components, the model including at least one user-configurable post component representing an event, the post component having at least one input port for receiving at least one input signal, the post component specifying a condition associated with the at least one input signal~~that is satisfied during execution of the model before a posting of a notice of said event occurs;~~

identifying when said condition is satisfied during the execution of said model;

posting said notice of an occurrence of said event in said modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event;

interrupting execution of an executing event in response to the determination of the occurrence of said specified event; and

performing an operation in said model in response to the determination of the occurrence of the specified event.

20. (Previously Presented) The method of claim 19 wherein said specified event is treated as a normal event and further comprising:

resuming execution of the interrupted event.

21. (Previously Presented) The method of claim 19 wherein said specified event is treated as an exception event and further comprising:

returning control of the execution of the model to a calling process which called the interrupted executing event without resuming execution of said interrupted event.

22. (Original) The method of claim 19 wherein said specified event is specified using an instantiated event object.

23. (Original) The method of claim 22 wherein said event is an explicit event.

24. (Original) The method of claim 22 wherein said event is an implicit event.

25. (Original) The method of claim 22 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

26. (Original) The method of claim 25 wherein said task object has at least one of a specified execution rate and priority.

27. (Previously Presented) The method of claim 26 wherein at least two events with different tasks are executing in a model and further comprising:

using event transition components to schedule the execution of components associated with said at least two events, said event transition components separating the execution of said components associated with said at least two events.

28. (Original) The method of claim 19 wherein the operation is controlled by an order of execution indicated in a branch priority block.

29. (Original) The method of claim 19 wherein the operation is the execution of more than one block group, said block groups being a user selected grouping of blocks, the order of execution of the block groups specified by a user.

30-32. (Canceled)

33. (Currently Amended) A physical computer-readable medium holding computer-executable instructions for controlling model execution in a graphical modeling environment, the instructions comprising:

one or more instructions for displaying a view of a graphical model with a plurality of executable time-based components, said graphical model including at least one user-configurable graphical post component representing an event, the graphical post component having at least one input port for receiving at least one input signal, the graphical post component specifying a condition associated with the at least one input signal~~that is satisfied during execution of said graphical model before a posting of a notice of said event occurs;~~

one or more instructions for identifying when said condition is satisfied during the execution of said graphical model;

one or more instructions for posting said notice of an occurrence of said event in said modeling environment to an event handler, said posting notifying said event handler of said occurrence of said event; and

one or more instructions for executing at least one component from said plurality of components in response to said notifying.

34. (Previously Presented) The medium of claim 33, wherein the instructions further comprise:

one or more instructions for registering at least one of said plurality of components with said event handler; and

one or more instructions for receiving at the at least one of said plurality of components registering with said event handler notification of the occurrence of said event following said posting.

35. (Currently Amended) The medium of claim 33, wherein ~~wherein~~ the graphical post component is a block or label.

36. (Currently Amended) The medium of claim 33, wherein the instructions further comprise :

one or more instructions for setting a sample time for an initial execution of at least one component to be the occurrence of the specified event;

37. (Previously Presented) The medium of claim 36, wherein the instructions further comprise:

one or more instructions for propagating the sample time to at least one other component in said graphical model, said at least one other component configured to inherit a sample rate.

38. (Previously Presented) The medium of claim 36, wherein the instructions further comprise:  
one or more instructions for setting a sample time of a plurality of non-contiguous components in said graphical model to be the occurrence of said event.

39. (Original) The medium of claim 38 wherein said sample time for the plurality of non-contiguous components is set without adjusting visible connections between components displayed in said view.

40. (Previously Presented) The medium of claim 36, wherein the instructions further comprise:  
one or more instructions for indicating with an event ID in said view that the sample time of said at least one component is set to said event.

41. (Original) The medium of claim 36 wherein said event is an explicit event set by a user.

42. (Previously Presented) The medium of claim 36 wherein said event is an implicit event caused by the execution of the graphical model.

43. (Original) The medium of claim 42 wherein the implicit event is one of power-up, power-down and initialization.

44. (Original) The medium of claim 42 wherein the implicit event corresponds to one of the enabling and disabling of a subsystem.

45. (Previously Presented) The medium of claim 34, wherein the instructions further comprise:  
one or more instructions for indicating which event a component receives with a user-configurable color in said view.

46. (Previously Presented) The medium of claim 33, wherein an execution scope of the specified event for which the execution of the graphical model is being monitored is restricted to a portion of the graphical model.

47. (Previously Presented) The medium of claim 33 wherein each event in said graphical model maps on a one-to-one basis to an event handler, said event handler being a function.

48. (Original) The medium of claim 47 wherein said function is inlined.

49. (Original) The medium of claim 33 wherein a branch priority block indicates an order of execution among at least two branches of blocks in response to said notifying.

50. (Original) The medium of claim 33 wherein more than one block group executes in response to said notifying, said block groups being a user selected grouping of blocks, the order of execution of the block groups specified by a user.

51. (Currently Amended) A physical computer-readable medium holding computer-executable instructions for controlling model execution, the instructions comprising:

one or more instructions for displaying a view of a model with a plurality of executable components, the model including at least one user-configurable post component representing an event, the post component having at least one input port for receiving at least one input signal, the post component specifying a condition associated with the at least one input signal that is  
~~satisfied during execution of said model before a posting of a notice of said event occurs;~~

one or more instructions for identifying the satisfaction of said specified condition during the execution of said model;

one or more instructions for posting the notice of an occurrence of said event in said modeling environment to an event handler, said posting notifying said event handler of the occurrence of said event;

one or more instructions for interrupting execution of an executing event in response to the determination of the occurrence of said specified event;; and

one or more instructions for performing an operation in said model in response to the determination of the occurrence of the specified event.

52. (Previously Presented) The medium of claim 51 wherein said specified event is treated as a normal event and wherein the instructions further comprise:

one or more instructions for resuming execution of the interrupted event.

53. (Previously Presented) The medium of claim 51 wherein said specified event is treated as an exception event and wherein the instructions further comprise:

one or more instructions for returning control of the execution of the model to a calling process which called the interrupted executing event without resuming execution of said interrupted event.

54. (Original) The medium of claim 51 wherein said specified event is specified using an instantiated event object.

55. (Original) The medium of claim 54 wherein said event is an explicit event.

56. (Original) The medium of claim 54 wherein said event is an implicit event.

57. (Original) The medium of claim 54 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

58. (Original) The medium of claim 57 wherein said task object has at least one of a specified execution rate and priority.

59. (Previously Presented) The medium of claim 58 wherein at least two events with different tasks are executing in a model and wherein the instructions further comprise:

one or more instructions for using event transition components to schedule the execution of components associated with said at least two events, said event transition components separating the execution of said components associated with said at least two events.

60. (Original) The medium of claim 51 wherein the operation is controlled by an order of execution indicated a branch priority block.



61. (Original) The medium of claim 51 wherein the operation is the execution of more than one block group, said block groups being a user selected grouping of blocks, the order of execution of the block groups specified by a user.